A General Technique to Measure Gradual Properties of Fuzzy Sets

Patrick Bosc
IRISA/ENSSAT
Technopole anticipa BP 80518
22305 Lannion Cedex France
patrick.bosc@enssat.fr

Ludovic Liétard
IRISA/IUT
Rue Edouard Branly BP 150
22302 Lannion Cedex France
ludovic.lietard@univ-rennes1.fr

Abstract

This paper considers gradual set-oriented functions and their application to fuzzy sets. An example, taken in the context of flexible querying of databases, is provided by the condition "the average salary of *young* employees is *around $4000*". The evaluation of this condition leads to apply the gradual set-oriented property "the average salary is *around $4000*" to the fuzzy set made of *young* employees. The contribution of this paper is to propose a general technique to achieve such computations which is a generalization of Sugeno fuzzy integral. The result is given in the form of a degree which has a clear meaning. Two examples of use of this technique in the area of regular relational databases serve to illustrate this proposal.

Keywords: relational database, flexible querying, linguistic summary, gradual set-oriented function.

1 Introduction

This paper deals with the problem of determining the value of a gradual set-oriented function when its arguments are fuzzy sets. Such a computation may appear into many contexts such as flexible querying of relational databases. In this context, atomic conditions define preferences instead of strict requirements and the set of answers returned to the user is discriminated. Atomic conditions are defined by fuzzy sets and are then called "vague" or "fuzzy" predicates. Such vague conditions can be aggregated using various operators (e.g., generalized conjunctions and disjunctions) and an extension of the SQL query language (called SQLf) has been proposed [1].

In SQLf, as well as in SQL, it is possible to consider aggregates (such as the cardinality, the maximum or the average) which are functions applying to a set of items. Aggregates can be integrated into flexible queries as: "retrieve the firms where the average salary is *around $2000*". Its expression in SQLf is:

> **select** #firm **from** EMP **group by** #firm
> **having** avg(salary) = *around(2000)*,

assuming that relation EMP(#emp, #firm, salary, age) describes employees working in different firms. The fuzzy condition "the average salary is *around $2000*" defines a gradual set-oriented function (which applies to sets of employees) and the satisfaction of each firm is given by the value of that function on the set of its employees. The more the firm satisfies this fuzzy condition, the higher its degree in the answer to the query.

In the previous example, we are in a simple situation where the gradual function can be computed since it applies to a crisp set. However, when the items to aggregate are issued from a fuzzy condition, the interpretation is no longer trivial, as in the query aiming at the retrieval of firms where "the average of *high* salaries is *around $2000*" which could be expressed in SQLf as:

> **select** #firm **from** EMP **where** salary = *high*
> **group by** #firm
> **having avg**(salary) = *around(2000)*.

Here, for a given firm, the gradual set-oriented function (which computes the extent to which the *average* salary is *around $2000*) applies to the fuzzy set of employees having a *high* salary.

The objective of this paper is to propose a technique to determine the value of a gradual set-oriented function when its arguments are fuzzy sets. No particular assumption is made about both the function and the fuzzy sets, and the function may have several arguments (each of them being a set). The result of the function is given in the form of a degree.

In the following, the gradual set-oriented function is denoted by F and A its argument (a fuzzy set of universe X). In case of a function F having several arguments, its fuzzy arguments are denoted by A^1, A^2, …, A^n. With these notations, the condition appearing in the previous example is rewritten F(A) where A is the fuzzy set of employees having a *high* salary whereas F is such that F(E) delivers the extent to which "the *average* salary of a crisp set E of employees is *around $2000*".

The rest of the paper is structured as follows. Section 2 proposes an approach to determine the value of $F(A^1, A^2, …, A^n)$. It is shown that his approach is a generalization of Sugeno fuzzy integral. Two examples of use of this proposition, respectively to define flexible queries in the context of regular relational database and to evaluate linguistic summaries, are introduced in section 3.

2 The proposed approach

The approach advocated to compute $F(A^1, A^2, \ldots, A^n)$ is introduced in subsection 2.1. In subsection 2.2, it is shown that this proposition is a generalization of Sugeno fuzzy integral.

2.1 A definition for $F(A^1, A^2, \ldots, A^n)$

According to our proposition, each fuzzy set A^i is viewed as a collection of its α-cuts which represent different interpretations (at different levels) of A^i. For a given level α, the value of $F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha)$ can be computed which is an interpretation (associated with level α) of the result $F(A^1, A^2, \ldots, A^n)$.

The question now is about the integration of the results associated with these various interpretations. Intuitively, it seems reasonable to think that :*the more often $F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha)$ is high when α varies, the higher $F(A^1, A^2, \ldots, A^n)$.* This interpretation is the meaning adopted for $F(A^1, A^2, \ldots, A^n)$.

A way is to look for the highest degree of satisfaction β such that, for each level α, $F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha)$ is at least equal to β. In other words:

$$F(A^1, A^2, \ldots, A^n) = \max_{\beta \text{ in } [0,1]} \min(\beta, \text{each}(\beta)), \quad (1)$$

where each(β) means "for each interpretation α, $F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha) \geq \beta$".

Obviously, this definition depends on that of each(β). The simplest case is when each(β) is Boolean and equals 1 as soon as each interpretation reaches the threshold β:

$$\text{each}(\beta) = \begin{cases} 1 \text{ if } \forall\alpha \text{ where } F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha) \text{ is defined} : F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha) \geq \beta \\ 0 \text{ otherwise.} \end{cases}$$

It is easy to show that this expression defines the truth value of $F(A^1, A^2, \ldots, A^n)$ as the minimum among all values $F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha)$ and expression (1) becomes:

$$F(A^1, A^2, \ldots, A^n) = \min_{\alpha \mid F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha) \text{ is defined}} F(A^1_\alpha, \ldots, A^n_\alpha).$$

This Boolean interpretation for each(β) may be too strict and a definition delivering a degree could be more convenient. We propose to sum the lengths of the intervals (of levels) where the threshold β is reached :

$$\text{each}(\beta) = \sum_{\substack{]\alpha_i,\alpha_j] \text{ such that } \forall\alpha\in]\alpha_i,\alpha_j], \\ F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha) \text{ is defined and } F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha) \geq \beta}} (\alpha_j - \alpha_i). \quad (2)$$

The higher each(β), the more numerous the levels α for which $F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha) \geq \beta$. In particular, each(β) equals 1 means that for each level α, $F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha)$ is larger than (or equal to) β.

This second definition for each(β) (i.e., expression (2)) is used in the rest of the paper.

In addition, from a computational point of view, the definition of $F(A^1, A^2, \ldots, A^n)$ (expression (1)) needs to handle an infinity of values β. However, it is possible to restrict computations to β values belonging to the set of "effective" $F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha)$ values :

$$F(A^1, A^2, \ldots, A^n) = \max_{\beta \in D} \min(\beta, \text{each}(\beta)), \quad (3)$$

where $D = \{\beta \mid \exists\alpha \text{ with } \beta = F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha)\}$.

Proof. We consider expression (1) and we show that a value λ out of D can be omitted in the computations. When λ is out of D, two cases can be considered:

Case 1. When λ is larger than the maximum value of D, we get each(λ) = 0 and consequently min(λ, each(λ)) = 0. Since the maximum value is retained in formula (1), value λ can be discarded since its contribution is 0.

Case 2. When λ is not larger than the maximum value of D, there are values of $F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha)$ such that $\lambda \leq F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha)$. We denote by m (associated with level α') the smallest one. We have: each(λ) = each(m) and λ ≤ m, and we get: min(λ, each(λ)) ≤ min(m, each(m)).

Since the maximum value is retained to define $F(A^1, A^2, \ldots, A^n)$, such a value λ can be omitted in computations of expression (1)♦

2.2 Position with respect to Sugeno fuzzy integral

This section points out that the previous approach is a generalization of Sugeno fuzzy integral. A particular property of F(A) is also provided when F is increasing with respect to set inclusion (this property will be used in subsection 3.2).

When F has one argument and is increasing with respect to set inclusion, expression (3) can be defined by:

$F(A) = \max_{\alpha \mid F(A_\alpha) \text{ is defined}} \min(F(A_\alpha), \alpha)$ (4).

When F is a fuzzy measure, expression (4) is that of a Sugeno fuzzy integral [2].

Proof. Since expression (3) deals with values in $\{\beta \mid \exists\alpha \text{ with } \beta = F(A_\alpha)\}$, F(A) can be rewritten :

$F(A) =$
$\max_{\alpha \mid F(A_\alpha) \text{ is defined}} \min(F(A_\alpha), \text{each}(F(A_\alpha)))$.

In addition, since F is increasing with respect to set inclusion: $\alpha \le \alpha' \Rightarrow F(A_\alpha) \ge F(A_{\alpha'})$. As a consequence, each($F(A_\alpha)$) is the maximum level λ where $F(A_\lambda) = F(A_\alpha)$. Consequently:

$F(A) = \max_{\alpha \text{ where } F(A_\alpha) \text{ is defined}} \min(F(A_\alpha), \chi(\alpha))$,

where $\chi(\alpha) = \max\{ \lambda \mid F(A_\lambda) = F(A_\alpha)\}$. Since each $\chi(\alpha)$ value is a level of α-cut and since F(A) is the maximum of $\min(F(A_\alpha), \chi(\alpha))$, we get:

$F(A) = \max_{\alpha \text{ where } F(A_\alpha) \text{ is defined}} \min(F(A_\alpha), \alpha)$. ♦

Property 1. When F satisfies the following constraints : i) it has a single argument, ii) it is defined for each α-cut of its argument, iii) it is increasing with respect to set inclusion, then we get:

F(A) is the maximum value δ such that $\delta \le F(A_\delta)$.

Proof. Let us denote by ϖ the α value which maximizes expression (4). We have $F(A) = \min(F(A_\omega), \varpi)$. Since $F(A) \le \varpi$ and $F(A) \le F(A_\omega)$, we get $F(A) \le F(A_\omega) \le F(A_{F(A)})$ (F being increasing). F(A) is the highest value having this property because if there exists λ such that $F(A) < \lambda$ and $\lambda \le F(A_\lambda)$ then $\min(\lambda, F(A_\lambda)) > F(A)$ which is impossible (the result of (4) being F(A)). ♦

3 Examples

This section illustrates the use of this proposal with two examples where it applies. The first example (subsection 3.1) is related to flexible querying of relational databases, the second one (subsection 3.2) is related to the evaluation of linguistic summaries.

3.1 Example in flexible querying

In this subsection we consider the use of aggregates in flexible querying of relational databases. We show that the approach introduced before allows to evaluate two types of condition involving aggregates.

Conditions of the first type are denoted by "agg (A) is C" where agg is an aggregate (maximum, average, …), A is a fuzzy set where the aggregate applies and C a fuzzy predicate. Such a condition expresses that the aggregate agg computed on fuzzy set A satisfies fuzzy condition C. An example is given by "the average salary of *high* salaries is *around $5000*" where agg is the average, A the fuzzy set of *high* salaries and C the fuzzy condition *to be around $5000.*

Conditions of the second type involve a fuzzy comparator θ and are denoted by "$agg_1(A)\ \theta\ agg_2(B)$". Such a condition expresses that the aggregate agg_1 computed on fuzzy set A is in relation θ with the aggregate agg_2 computed on fuzzy set B. An example is given by "the average of *high* salaries is *almost equal to* the maximum salary of *medium* salaries" where agg_1 is the average, A the fuzzy set of *high* salaries, θ the fuzzy comparator *almost equal to*, agg_2 the maximum and B the fuzzy set of *medium* salaries.

The evaluation of a condition "agg(A) is C" leads to determine the value of the gradual set-oriented function "the aggregate is C" on fuzzy set A. If F is the function defined by:

$$F : 2^X \to [0,1]$$
$$E \to F(E) = \mu_C(agg(E)),$$

(where X is the universe of A) the evaluation of "agg(A) is C" leads to compute F(A). According to our approach, the more α-cuts of A highly satisfy the constraint "the aggregate is C", the more satisfied "agg(A) is C".

Example 1. The statement "avg(A) is *high*" is considered with the following fuzzy set A:

$\{0.1/1 + 0.1/2 + 0.1/3 + 0.1/4 + 0.1/5 + 0.1/15 + 0.2/200 + 0.5/700 + 0.8/500 + 1/600\}$.

Function F is defined by $F(E) = \mu_{high}(avg(E))$ and the interpretation of "avg(A) is *high*" needs to compute F on A. If we assume that: $\mu_{high}(203) = 0.2$, $\mu_{high}(500) = 0.8$, $\mu_{high}(550) = 0.9$ and $\mu_{high}(600) = 1$, we get :

α	0.1	0.2	0.5	0.8	1
$avg(A_\alpha)$	203	500	600	550	600
$F(A_\alpha)$	0.2	0.8	1	0.9	1

Table 1. The truth values of $F(A_\alpha)$

From the previous table we get :

$D = \{0.2, 0.8, 0.9, 1\}$.

Figure 1 gives: each(0.2) = 1, each(0.8) = 0.9, each(0.9) = 0.8 and each(1) = 0.5.

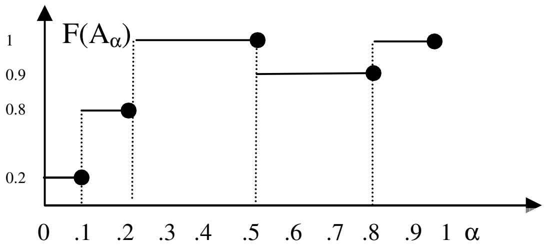

Figure 1. Values for $F(A_\alpha)$

According to (3), the result is:

$$F(A) = \max \min(0.2, 1) \min(0.8, 0.9) \min(0.9, 0.8) \min(1, 0.5) = 0.8.$$

The statement "avg(A) is *high*" is rather true (at degree 0.8) since every interpretation of fuzzy set A (except one) strongly satisfies condition "the average is *high*" (cf. figure 1).♦

The evaluation of conditions of type "$agg_1(A)\ \theta\ agg_2(B)$" leads to define a gradual set-oriented function expressing that, for two crisp sets E_1 and E_2, $agg_1(E_1)$ is in relation θ with $agg_2(E_2)$. If X_1 is the universe where A is defined, and X_2 that of B we get :

$$F : 2^{X_1} \times 2^{X_2} \to [0,1]$$
$$(E_1, E_2) \to F(E_1, E_2) = \mu_\theta(agg_1(E_1), agg_2(E_2)).$$

The evaluation of "$agg_1(A)\ \theta\ agg_2(B)$" is given by the computation of F(A, B). According to our approach, the more the α-cuts of A and B highly satisfy the constraint "$agg_1(A_\alpha)\ \theta\ agg_2(B_\alpha)$", the more satisfied "$agg_1(A)\ \theta\ agg_2(B)$".

Example 2. We consider the condition "the average salary of *young* employees is *almost equal to* the maximum of *medium* salaries" where the set of salaries tied to *young* employees is :

$$A = \{1/3000 + 0.8/10\,000 + 0.6/2000 + 0.3/11\,000 + 0.1/4000\},$$

while the set of *medium* salaries is :

$$B = \{0.2/6000 + 0.5/4800 + 0.8/5800 + 0.1/1000\}.$$

Let us consider the fuzzy comparison operator *almost equal to* defined as:

$$\mu_\approx(x, y) = 1 - \min(|x - y|/1000, 1),$$

where x and y belong to the set S of salaries. The evaluation of "avg(A) ≈ max(B)" leads to compute F(A, B) where function F is defined for crisp sets :

$$F : 2^S \times 2^S \to [0,1]$$
$$(E_1, E_2) \to F(E_1, E_2) = \mu_\approx(avg(E_1), max(E_2)).$$

The values of F for the effective α-cuts are given by table 2 and figure 2:

α	avg(A_α)	max(B_α)	F(A_α, B_α)
0.1	6000	6000	1
0.2	6500	6000	0.5
0.3	6500	5800	0.3
0.5	5000	5800	0.2
0.6	5000	5800	0.2
0.8	6500	5800	0.3
1	3000	not defined	

Table 2. Values for $F(A_\alpha)$

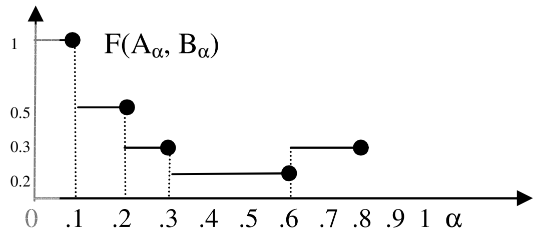

Figure 2. Values for $F(A_\alpha, B_\alpha)$.

The set D concerned by definition (3) is then D = {0.2, 0.3, 0.5, 1}, and, from figure 2, we compute each(0.2) = 0.8, each(0.3) = 0.5, each(0.5) = 0.2, each(1) = 0.1. From definition (3) we compute :

F(A, B) = max min(1, 0.1), min(0.5, 0.2), min(0.3, 0.5), min(0.2, 0.8) = 0.3.

This result is rather low because for most of α-cuts, the satisfaction of "avg(A_α) ≈ max(B_α)" is rather low (cf. figure 2).♦

3.2 Evaluation of linguistic summaries

Database knowledge discovery aims at discovering hidden knowledge or patterns from databases. Many items can be the subject of discovery, among them are linguistic summaries of data which are statements of the natural language.

Linguistic summaries involving linguistic quantifiers [3, 4, 5] are expressing a constraint on the cardinality (or proportion) of items satisfying a fuzzy condition. It is the case of the linguistic summary "*most of* employees are *young*" where the proportion of *young* employees (with respect of the entire database) is in accordance with the linguistic quantifier *most of*. Such a summary is associated with a degree expressing its validity. Linguistic summaries can also express a property between two concepts [6, 7] as in "*young* employees are *well-paid*". This summary can have a Boolean interpretation [6] stating that: for each employee of the database, "the *younger* he is, the more *well-paid* he is".

In this section, we consider a relational database and a new type of linguistic summaries. More precisely, we consider linguistic summaries of the form "A^1 is C^1 and A^2 is C^2 and … and A^n is C^n" where A^is are attributes of a same relation and C^is are different linguistic variables. Each linguistic variable C^i is defined by a fuzzy set over the domain of attribute A^i. An example is provided by the linguistic summary "the employees are *young* and *well-paid*" which is rewritten "age is *young* and *salary* is well-paid" where age and salary are two attributes of the relation R describing employees.

The validity of such a summary over the entire database depends on two parameters : the quantity of tuples which satisfy the conjunction and their levels of satisfaction. The higher are the quantity and the level of satisfaction, the more valid the summary is.

The difficulty of this interpretation is due to the fact that, in general, the higher the quantity, the smaller the satisfaction. This means that the interpretation needs to determine a compromise between these two aspects, quantity and individual satisfaction. In this section, we show that such a compromise can be achieved by the approach presented in the previous section. The result can be easily interpreted by the user depending on the definition he gave for the C^is.

Let F be the gradual set-oriented function defined by :

$$2^{\mathrm{dom}(A_1) \times \ldots \times \mathrm{dom}(A_n)} \rightarrow [0,1]$$
$$E \rightarrow F(E),$$

where F(E) expresses the percentage of elements from $R[A^1, ..., A^n]$ which can be found in E :

$$F(E) = |E \cap R[A^1, ..., A^n]| \,/\, |R[A^1, ..., A^n]|.$$

The value of $F(C^1 \times C^2 \times \ldots \times C^n)$ estimates the percentage of elements from R which satisfies "A^1 is C^1 and A^2 is C^2 and … and A^n is C^n". F being increasing and defined for each α-cut, from expression (4) we get :

$$F(CP) = \max_{\alpha\in[0,1]} \min(F(CP_\alpha), \alpha),$$

where CP is the Cartesian product $C^1 \times C^2 \times \ldots \times C^n$. As the α-cut of a Cartesian product is the Cartesian product of α-cuts we get :

$$F(C^1 \times C^2 \times \ldots \times C^n) = \max_{\alpha\in[0,1]} \min(F(C^1{}_\alpha \times C^2{}_\alpha \times \ldots \times C^n{}_\alpha), \alpha) \qquad (5)$$

According to property 1, $F(C^1 \times C^2 \times \ldots \times C^n)$ is the maximum value p such that $p \leq F(C^1{}_p \times C^2{}_p \times \ldots \times C^n{}_p)$. In other words, it delivers the highest p such that at least p% of the tuples satisfies "A_1 is C_1 and A_2 is C_2 and … and A_n is C_n" at a degree which is of at least p.

P = 1 means that all the database fully satisfies "A^1 is C^1 and A^2 is C^2 and … and A^n is C^n". P = 0.8 means that at least 80% of the database satisfies "A^1 is C^1 and A^2 is C^2 and … and A^n is C^n" at a degree at least equal to 0.8. When p is small (0.1 for example), this summary is not interesting, since p is a maximum.

From a computational point of view, two points are worthy of discussion :

i) definition (5) needs to handle an infinity of values α and next proof shows that it can be limited to values belonging to the following set :

$$D' = \{\min(\mu_{C^1}(t.A_1), \mu_{C^2}(t.A_2), \ldots, \mu_{C^n}(t.A_n)), t \in R\},$$

ii) the computation of $F(C^1{}_\alpha \times C^2{}_\alpha \times \ldots \times C^n{}_\alpha)$ involves the intersection between a Cartesian product of α-cuts and a projection :

$$(C^1{}_\alpha \times C^2{}_\alpha \times \ldots \times C^n{}_\alpha) \cap R[A^1, ..., A^n].$$

This computation may appear to be difficult in practice since set $(C^1{}_\alpha \times C^2{}_\alpha \times \ldots \times C^n{}_\alpha)$ is (most of the times) infinite. However, no particular difficulties are expected here since $C^1{}_\alpha \times C^2{}_\alpha \times \ldots \times C^n{}_\alpha$ is a crisp set which means that this intersection can be obtained by a projection and a Boolean selection over relation R.

Proof. We consider expression (5) and we show that a value λ out of D' can be omitted in the computations. When λ is out of D', two cases can be considered :

Case 1. When λ is larger than the maximum value of D', no values from $R[A^1, ..., A^n]$ belong to $F(C^1{}_\lambda \times C^2{}_\lambda \times \ldots \times C^n{}_\lambda)$. As a consequence we get $F(C^1{}_\lambda \times C^2{}_\lambda \times \ldots \times C^n{}_\lambda) = 0$. Since the maximum value is retained in formula (5), value λ can be discarded since its contribution is 0.

Case 2. When λ is not larger than the maximum value of D', there are values p of D' such that $\lambda \leq p$. We denote by m the smallest one. We have:

$F(C^1{}_\lambda \times C^2{}_\lambda \times \ldots \times C^n{}_\lambda) = F(C^1{}_m \times C^2{}_m \times \ldots \times C^n{}_m)$ and $\lambda \leq m$,

and we get:

$$\min(F(C^1{}_\lambda \times C^2{}_\lambda \times \ldots \times C^n{}_\lambda),\lambda) \leq \min(F(C^1{}_m \times C^2{}_m \times \ldots \times C^n{}_m),m).$$

Since the maximum value is retained to define $F(C^1 \times C^2 \times \ldots \times C^n)$, such a value λ can be omitted in computations.♦

4 Conclusion

This paper has proposed an approach to determine the value of a gradual set-oriented function when its arguments are fuzzy sets. If the function is denoted by F while its fuzzy arguments are denoted by $A^1, A^2, \ldots, A^n$, the value of $F(A^1, A^2, \ldots, A^n)$ is a degree such that the more often $F(A^1_\alpha, A^2_\alpha, \ldots, A^n_\alpha)$ is high when α varies, the higher $F(A^1, A^2, \ldots, A^n)$. In the particular case where F is a fuzzy measure, the proposed computation is a Sugeno fuzzy integral. Two examples of use of this proposition have been provided.

The first one concerns the evaluation of conditions involving aggregate operators in flexible querying of regular databases. When considering a condition of type "agg(A) is C" (avg(*high* salaries) is *around 2000$*) the given interpretation considers that the more α-cuts of A highly satisfy the constraint "the aggregate is C", the more satisfied "agg(A) is C".

In addition, it can be demonstrated [8] that the truth value obtained for "agg(A) is C" is the negation of that of "agg(A) is not C" only when A is a normalized fuzzy set or the aggregrate is defined for the empty set. Otherwise, this property does not hold and the truth value of these two statements is bounded by the maximum membership in A, which indeed reflects the inapplicability of the aggregate on the empty α-cuts. Conditions of type "$agg_1(A)\ \theta\ agg_2(B)$" ("$avg(A) \approx max(B)$") where θ is a fuzzy comparator, can also be interpreted and, here again, the more α-cuts of A and B highly satisfy "$agg_1(A_\alpha)\ \theta\ agg_2(B_\alpha)$", the more satisfied "$agg_1(A)\ \theta\ agg_2(B)$".

The second use of our approach is related to the evaluation of linguistic summaries of the type "A^1 is C^1 and A^2 is C^2 and … and A^n is C^n" where A^is are attributes of a same relation and C^is are different linguistic variables. The validity of such a summary over the entire database depends on two parameters : the quantity of tuples which satisfies the conjunct and their individual levels of satisfaction. As, in general, the higher the quantity, the smaller satisfaction, our proposition determines a compromise between these two aspects. It delivers the highest p such that at least p% of the tuples satisfies "A^1 is C^1 and A^2 is C^2 and … and A^n is C^n" at a degree which is of at least p.

In the near future, we aim at designing new patterns for linguistic summaries where the conjunctions are replaced by disjunctions or are mixed with other logical operators. In addition, the design of efficient algorithms to achieve the proposed computations is also a matter of future research.

References

[1] P. Bosc, O. Pivert (1995), SQLf: A relational database language for fuzzy querying. *IEEE Transactions on Fuzzy Systems,* vol. 3, pages 1-17.

[2] M. Sugeno (1974), *Theory of fuzzy integrals and its applications*, Ph. D. Thesis, Tokyo Institute of Technology, Tokyo (Japan).

[3] R.R. Yager, M. Ford, A.J. Canas (1990), An approach to the linguistic summarization of data. In *Uncertainty in Knowledge Bases,* (B. Bouchon-Meunier, R.R. Yager and L.A. Zadeh eds), Lecture Notes in Computer Science 521, Springer-Verlag, pages 456-468.

[4] D. Rasmussen, R.R. Yager (1996), Summary SQL – A flexible fuzzy query language. Proceedings of the *Flexible Query-Answering Systems Workshop (FQAS'96),* Roskilde (Denmark), pages 1-18.

[5] R.R. Yager (1996), Database discovery using fuzzy sets. *International Journal of Intelligent Systems,* vol. 11(9), pages 691-712.

[6] P. Bosc, O. Pivert, L. Ughetto (1999), Database mining for the discovery of extended functional dependencies. Proceedings of the *18th North American Fuzzy Information Processing Society Conference (NAFIPS'96),* New-York City (USA), pages 580-584.

[7] P. Bosc, D. Dubois, O. Pivert, H. Prade, M. de Calmès (2002), Fuzzy summarization of data using fuzzy cardinalities. Proceedings of the *9th International Conference on Information Processing and Management of Uncertainty (IPMU'2002),* Annecy (France), pages 1553-1559.

[8] P. Bosc, L. Liétard (2004), Non monotonic aggregates applying to fuzzy sets in flexible querying. Proceedings of the *10th International Conference on Information Processing and Management of Uncertainty (IPMU'2004),* Perugia (Italy), pages 11-18.